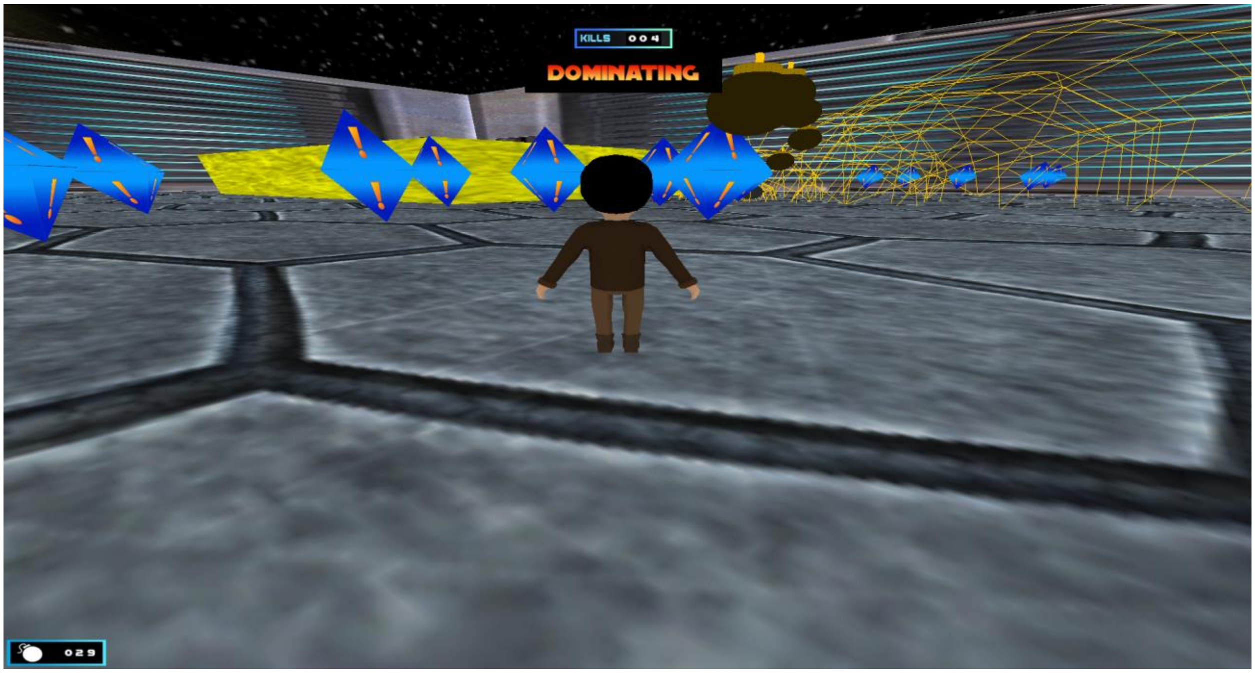
AETHER BATTLE
PLAYER'S GUIDE

Name of Game: Aether Battle

Project: CSC165 Assignment #3: Final Game Project

Author: Tan Dao

Screenshot In-Game:



How to Compile & Run from Command Line:

Launch the game by running the 'run.bat' file in the 'A3 Game Client' folder. This will cause a pop-up window to display where you can choose whether you want to play in 'Single Player' or 'Multi-Player' mode, as well as select whether you prefer Windowed or Full Screen mode. If you would like to play with multiple players, you first need to launch the server program by running the 'run.bat' file in the 'A3 Game Server' folder. It will prompt you to select a port number to host your game on. It's advised that you pick a number over 1000. After that, *then* you can launch the clients and have them start the game in 'Multi-Player' mode. Be sure to fill in empty boxes with the appropriate Server IP Address and Server Port number.

If you'd like to run the game from the command line and not use the run.bat files, follow the following instructions:

Open the command line and change the directory into the folder containing the submitted material:

```
cd [insert directory address here]
```

Copy and paste the following lines into the command prompt to compile and run:

```
javac -d bin -sourcepath src src\a2\Starter.java

cd bin

java -Dsun.java2d.d3d=false a2.Starter
```

Hotkeys:

NOTE The game supports the use of a Keyboard and Gamepad (a Logitech F310 GamePad was used during development), but the game will still run without a Gamepad attached.

Player 1 (Keyboard):

'SPACEBAR': Toss a Bomb

'W': Move FORWARD

'S': Move BACKWARD

'A': Move LEFT (Strafe left)

'D': Move RIGHT (Strafe right)

'Q': Tilt avatar's heading toward the LEFT

'E': Tilt the avatar's heading toward the RIGHT

'UP Arrow': Raise camera's elevation

'DOWN Arrow': Lower camera's elevation

'LEFT Arrow': Turn camera view towards the LEFT

'**RIGHT Arrow':** Turn camera view towards the RIGHT

'ESCAPE': Close the game

Player 2 (GamePad):

'BACK' / 'START': Toss a Bomb

'Right Joystick': Orbits the camera around the player's avatar

'Y': Move FORWARD

'A': Move BACKWARD

'X': Move LEFT (Strafe left)

'B': Move RIGHT (Strafe right)

'LB': Tilt avatar's heading toward the LEFT

'RB': Tilt the avatar's heading toward the RIGHT

How to Play:

The object of the game is to obtain Bombs and throw them at your opponents. Players can't throw unlimited Bombs, but instead must pick up Loot Boxes scattered across the stage. Running your avatar over a Loot Box grants you +1 Bombs. However, if you're currently on a killing spree (killing multiple opponents in a row without dying yourself), your Bomb explosions will change color and picking up a Loot Box will grant you Bonus Bombs. Here's the breakdown**:**

Killing Spree	**Announcer Message**	**Bomb Explosion Color**	**Bombs earned per Loot Box**
3	KILLING SPREE	**Orange**	2
4	DOMINATING	**Orange**	2
5	MEGA KILL	**Orange**	2
6	UNSTOPPABLE	**Magenta**	3
7	WICKED SICK	**Magenta**	3
8	MONSTER KILL	**Magenta**	3
9	GODLIKE	**Pink**	4
10	HOLY SHIT	**Green**	5

There is an NPC roaming around the map that looks like a rectangle. It behaves similar to a Rooma vacuum, so I've named it the Doomba. Being hit by this will kill you instantly, ending your killing spree, and causing your Bomb count to be reduced to 0. On 'Single Player Mode', the Doomba can be killed, but on 'Multi-Player Mode', the Doomba is completely immune. Killing a hostile target will grant you +1 to your score. Dying to the Doomba will cause you to lose all of your Bombs, and dying to an enemy player will cut your Bomb count in half. Getting hit by an enemy player's Bomb will also instantly kill you.

Scripting:

Scripting was only used to initialize the 4 walls that surround the edge of the stage. The main 'AetherBattle.java' class obtains these objects and adds them to the gameworld. Aside from that, an end-user wouldn't be able to make any use of scripting.

Genre/Theme/Dimensionality/Activities

Genre: Action

Theme: Space

Dimensionality: 3D, 3rd person camera

Activities: Combat, item collecting, avoiding NPC

How requirements were satisfied:

External Models: The three models in the game include the avatar, the Loot Boxes, and the Bomb objects that the players can throw. All were made in Blender and imported using SAGE's built-in loader. All models were skinned using UV-unwrapping and are properly textured in-game.

Networked Multi-player: Multi-player support is implemented. A separate server program must be run prior to launching the individual clients to do so. Once in-game, all players can see each other, and their movements and rotations can be seen as they change in real time.

Scripting: Used to initialize the 4 walls surrounding the stage. Implemented using JavaScript. The .js file is titled 'boundaries.js' located in the src/a3 folder.

Skybox and Terrain: Both a Skybox and Terrain are implemented using the appropriate SAGE calls. However, the terrain is not visible to the player, since it was not intended that it have any significant impact on gameplay. In the AetherBattle.java file, the line that adds the Terrain to the gameworld is on Line 298. The Skybox is visible if you move the avatar to the edge of the stage and turn the 3rd person camera so that the outside of the arena can be seen.

Events: An event is triggered when a player runs their avatar into a Loot Box to gain more Bombs in their arsenal. Event handling via input devices is also dealt with in the initInput() function.

3D Sound: A plethora of sound effects are included in the game, including background music, Bomb toss sound, Bomb explosion sound, killing spree announcer voicelines, collecting Loot Boxes, and a sound effect upon dying.

HUD: the HUDImage class was utilized alongside Photoshop to display the two .png images that show the player's current Score and Bomb count. A separate HUDNumber class was written to allow for loading external images of individual digits using a different font.

Heirarchical SceneGraph: Upon loading the game, if you look up you can see a simple solar system Group that rotates every frame.

Animation: I was able to successfully keyframe and pose my model in Blender in such a way that it has a smooth and realistic walking animation. However, when I import the model into Blender and start the animation the avatar looks completely distorted (and almost horrifying enough to be in a Silent Hill game). I've spent many hours trying to remedy the problem, but have only managed to identify the core issue without being able to fix it. When SAGE loads the model, it doesn't import the scaling values for the armature (it sets the armature scaling factors to 1.0), and I used a factor less than 0.2.

To at least prove that I did the animation in Blender, I have included the Blender file where you can view the animation for yourself, but I opted not use to use it in the final iteration of the game (because it looks, you know, terrifying af).

Heirarchical SceneGraph: Upon loading the game, if you look up you can see a simple solar system Group that rotates every frame.

NPCs: An NPC is loaded in both Single Player and Multi-Player. Its AI is handled using a Behavior Tree. NPCs are updated locally, so each player sees a unique NPC that others can't see. The NPC moves in a straight line and randomly changes direction every 3 seconds. If it hits a wall it does a 180 degree turn.

Physics: The SAGE physics engine was integral for making the Bomb toss animation work. Every time a player throws a bomb a new spherical physicsobject is created and attached to the Bomb, and a force vector is applied to it to simulate the Bomb being throw from the player.

Things That Don’t Work:

As stated above, I was unable to get animation to load correctly from Blender into the game using SAGE. The sound effects don’t utilize 3D sound. No matter where the player is and where the source of the sound is, it’ll play at the same volume universally.

Another really big issue I’ve been struggling with during the last hurdle of development was object management to reduce render lag. After enough bombs are tossed, the game starts to take an FPS drop. This is a huge problem in multiplayer because a player isn’t confirmed as being hit unless they themselves see it happen on their end, even if another player saw it happen at full speed on their screen.

GamePad Used: Logitech F310 GamePad

http://gaming.logitech.com/en-us/product/f310-gamepad

Contributions from Team: I opted to do this project alone, so I’m fully responsible for everything submitted.

Items I Created Myself: I followed video tutorials to create the models used in-game.

Computer Used to Test Game: ‘Pac-Man’ & the one immediately to its LEFT (don’t remember the name) in RVR5029